Tripod Operators for Recognizing Objects in Range Images; Rapid Rejection of Library Objects.

Frank Pipitone
and
William Adams

Navy Center for Applied Research
in Artificial Intelligence,
Naval Research Laboratory,
Washington, DC 20375-5000

Abstract

The tripod operator is a class of feature extraction operators for range images which facilitate the recognition and localization of objects. It consists of three points in 3-space fixed at the vertices of an equilateral triangle and a procedure for making several scalar measurements in the coordinate frame of the triangle. The triangle is then moved as a rigid body until the three vertices lie on the surface of some range image or modeled object. The resulting measurements are local shape features which are invariant under rigid motions. These features can be used to automatically find distinctive regions at which to begin recognition, to rapidly screen candidate objects for a match, and to speed pruning in the generation of interpretation trees. Tripod operators are applicable to all 3-D shapes, and reduce the need for specialized feature detectors. A key property is that they can be moved on the surface of an object in only three DOF (like a surveyor's tripod on the ground). Consequently, only a 3-dimensional manifold of feature space points can be generated, for any dimensionality of feature vector. Thus, objects can be represented compactly, and in a form allowing fast matching.

They are used here to characterize objects by generating a cloud of points in feature space for each object by random placement of the operator. Then new feature measurements are made by operator placements in a range image containing one of those objects. Using a simple nearest-neighbor approach, we determine which objects are rejected and which remain as recognition candidates. Experiments were performed using this approach, showing that tripod operators have excellent discriminating power.

1. Introduction

In recent years, research in the acquisition and use of range images has led to the development of increasingly fast and accurate rangefinders [6] and to a variety of increasingly effective methods for recognizing and locating objects in range images. The fundamental limits of performance have, however, not nearly been reached. This paper pursues the goal of high speed object recognition by introducing a class of range image operators that extract local shape information that is invariant under rigid motions of the object with respect to the rangefinder. These operators can be applied to 3-D objects of any shape. They exploit the fact that a small number (e.g., four to eight) of range measurements often contain a large amount of information about the identity and pose of objects on which they lie, particularly when the range data is very precise. The operators arose from studying the problem of efficiently mapping small sets of range measurements into sets of possible poses of various candidate objects. This was achieved by structuring the range data so that the mapping involves sets small enough to compute offline and store.

The application of the tripod operator produces a numerical feature vector which retains much of the surface shape information contained in the range measurements involved, but no other information. Object pose can be recovered, if desired, by making use of the location of the operator in the coordinate frame of the rangefinder. We will describe here various properties and potential applications of tripod operators, concentrating on the problem of rapidly recognizing a single object selected from a library of objects which the system has seen before. We present experimental results showing that the tripod operator has very strong discriminating power. In many cases, a single operator application provides decisive evidence for the rejection of a candidate object, or strong evidence that an object does match.

The most closely related previous work involves the exploitation of geometric constraints to recognize rigid objects in range images. Grimson [1,5] extensively developed the idea of searching for associations between image features and model elements consistent with geometric constraints among the model elements, using *interpretation trees* to represent the consistent hypothesised associations (interpretations). This general approach was introduced by several authors [2,3,4,10] within a short time. Our work differs from these efforts in that we provide mechanisms for efficiently prestoring model information so that the costly early stages of interpretation tree generation can be avoided at recognition time. Lamdan and Wolfson [11] provide for such precompilation in a wide variety of vision problems, but require the existence of a reasonably small number of reasonably stable *interest points*, whereas our operators are to be used

anywhere on a surface. In contrast to [5], we use both dense range images and sparse sets automatically chosen from such images. In an initial report [12], we argue that the tripod operator should allow very fast recognition. Here we present experimental evidence supporting this.

2. Definition of the Tripod Operator

A tripod operator of order n consists of three points, called *feet*, at the vertices of a (usually equilateral) triangle having fixed prescribed edge lengths, and a procedure for making n scalar measurements of a surface in a coordinate frame determined by the triangle. A tripod operator is to be applied to a two-dimensional surface imbedded in three-space. This is generally in the form of a computer representation of a rigid physical object, such as a surface interpolation of a range image, or a surface model of an object obtained from a computer aided design system or from range images. The operator is applied by rotating and translating it as a rigid body until its three feet all lie on the surface, much as in placing a surveyor's transit or a camera tripod. Next, the n scalar measurements are made and regarded as a feature vector **f** of length n. **f** is an intrinsic property of the object represented by the surface; it depends on the shape of the object and on where the operator is placed on the object, but not on where the object and tripod are located in any coordinate system.

As an introductory example, consider the simple order 1 operator shown in Fig. 1a. It consists of feet A, B and C at the vertices of an equilateral triangle of edgelength d, and a "probe" line passing through the center of the triangle and perpendicular to its plane. Now if the probe line intersects the surface at a point denoted by D, the distance from D to the plane of the triangle is the feature value generated. If A, B, C, and D fall at position vectors $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$, respectively, then

$$s = \frac{((\mathbf{p}_2-\mathbf{p}_1)\times(\mathbf{p}_3-\mathbf{p}_1))\cdot((\mathbf{p}_3-\mathbf{p}_4)}{\|((\mathbf{p}_2-\mathbf{p}_1)\times(\mathbf{p}_3-\mathbf{p}_1))\|}$$

can be used to compute the value of this tripod feature. A positive feature value represents a local depression in the surface, and a negative value represents a local "bump". Note in Fig. 1b that we can generalize this operator to an order n operator by using n arbitrary space curves $\{\mathbf{x}_1(s_1), \mathbf{x}_2(s_2), ..., \mathbf{x}_n(s_n)\}$ which we call probe curves. Each probe curve $\mathbf{x}_i(s_i)$ is a position vector as a function of a scalar parameter s_i, which represents arc length along the curve. The application of the operator to a surface results in the values of the n scalars s_i determining where the probe curves intersect the surface.

3. Linkable Tripod Operators

We now describe a class of tripod operators with particularly interesting properties which allow them to be chained, or *linked* together (see section 4.1). We will start with an order 1 example. The three feet A, B, and C of the tripod are at the vertices of an equilateral triangle of edge length d, and a probe curve is formed by a circle centered at the midpoint of the edge BC and coaxial with it, as shown in Fig. 1c. The radius is $\sqrt{3}d/2$, so that any point D on the circle is at a distance d from both B and C. When applied to a surface, four point operator returns one parameter value, the angle θ between the triangles ABC and BDC, where D is a point where the circle intersects the surface. Our convention is that $\theta = 180^o$ for a planar surface, with $\theta > 180^o$ if the hinge edge BC looks convex from the rangefinder's viewpoint. The application of the operator to a surface, yields $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ as the position vectors of A, B, C and D, respectively, and the scalar feature θ. Note that this operator is simply two triangles hinged at a common edge. It can be generalized to an operator of order n by hinging together n+1 triangles and defining one of them as the tripod. For example, Fig. 1d shows an order 3 operator. Here points E and F are similar in function to D; after planting A, B and C on a surface, D, E and F are consecutively moved through their respective circular paths until they strike the surface, yielding three feature values θ_1 , θ_2 and θ_3. Fig. 1e shows an order 9 operator. The points are computed sequentially from A through L.

4. Two Uses of Tripod Operators

In a complete recognition/localization system, there is first a need to reject impossible candidate objects from the library of known objects as rapidly as possible. Doing this with tripod operators (section 6) is the main subject of the experimental work in this paper. At a later stage, if localization is desired, the system must bring to bear enough geometric information from the image to determine the pose of a final single candidate. This we approach by combining the ideas of interpretation tree search and tripod operators (section 4.1) For the latter, one requires object models tiled with M small surface patches.

4.1 Recognition and Localization by Linking Successive Operator Placements

Note that the operator of Fig. 1c has a certain symmetry; after it is applied to a surface, it makes little difference which triangle is regarded as the tripod. This leads to the idea of making a second application of the operator at the three points $\mathbf{p}_2$, $\mathbf{p}_4$ and $\mathbf{p}_3$ on the surface of a range image, producing a new point $\mathbf{p}_5$ as shown in Fig. 2 and a new feature value θ'. Thus for the second application of the operator, A, B, C and D are at $\mathbf{p}_2$, $\mathbf{p}_4$, $\mathbf{p}_3$ and $\mathbf{p}_5$, respectively. Let us now define a *k-interpretation* as an association of

k points $\mathbf{p}_i$ with k respective patches on which they might lie.

Now note that we have succeeded in linking these operators together, so that we can combine the information gotten from their feature values. If we use the first operator application to look up the prestored 4-interpretations of $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$ and $\mathbf{p}_4$ for some model, and the second to look up the 4-interpretations of $\mathbf{p}_2$, $\mathbf{p}_4$, $\mathbf{p}_3$ and $\mathbf{p}_5$, we can retain the 5-interpretations consistent with both. This linking procedure can be repeated indefinitely. Figure 2 shows five operator applications, yielding eight points and five feature values. This example illustrates the opportunistic growing of links wherever they don't cross boundaries of image segments. One good mechanism for keeping track of these sets of consistent interpretations is interpretation trees , with the range points $\mathbf{p}_i$ as the sensor measurements and the surface patch as the model elements, much as in [5]. The difference here is that the constraints among four measurements at a time are included at each new tree level, thus eliminating many branches without generating them. Also, the constraints are somewhat stronger taken among four points at once, since a 4-interpretation satisfying the six pairwise constraints separately might not satisfy them simultaneously, and the latter is enforced by the 4-point operator.

The linking could be done using one or two common points instead of three as described, but linking three points has the advantage of preserving rigidity; the distance between any two points in Fig. 2 is known to within the uncertainties arising from finite patch size and measurement error. For linkable operators of order greater than 1, the procedure generalizes in the obvious way; an outer triangle of one operator placement is used as the tripod of the next placement. We are currently planning experiments with linking procedures.

4.2 Fast Rejection of Candidates Using Isolated Operator Placements

In section 6 we will describe the results of experiments that show that single placements of a tripod operator can be highly discriminating between objects. The basic idea here is to preprocess each object in a library of objects by applying some tripod operator at many random locations on its surface. The resulting "cloud" of points in feature space is recorded for each object. We will give both theoretical (section 5.1) and experimental (section 6) evidence that the required number of operator placements to thoroughly characterize an object is manageable.

At recognition time, an operator placement is made on a range image of a some scene possibly containing objects that were preprocessed, producing a feature vector $\mathbf{f}$. For each prestored point cloud, a calculation is done to determine whether $\mathbf{f}$ is close enough to some point in the cloud to be possibly from that object. If so, the object

remains a candidate; otherwise the object is rejected. In section 5.1 we argue that these clouds are inherently sparse, occupying manifolds of dimensionality not exceeding three. This explains the strong discriminating power of operators of order 4 or greater observed in our experiments. Note that this approach does not require an explicit complete surface model, since the preprocessing is done on raw range images of objects.

This approach depends on the assumption that almost any new operator placement on object 1 will yield a feature space point closer than some threshold distance to the nearest point in the stored point cloud for object 1, and that a reasonable fraction of the points obtainable from object 1 are farther than this threshold from the object 2 cloud, for most pairs of objects of interest. This assumption is experimentally validated in section 6. This simple recognition logic could easily be somewhat improved by using a Bayesian statistical approach, but the present approach is simple, performs well, and illustrates the salient characteristics of this recognition problem.

In the above approach, if the scene contains more than one object, grouping, or segmentation, becomes an issue. As with previous approaches [1,10,11], the problem of avoiding being fooled by measurements lying on multiple objects is not insuperable. One can subject the range image to a segmentation procedure which results in the labeling of each pixel as a member of a region such that two pixels in the same region probably lie on different objects. One hopes to achieve this with as few as possible regions lying on any one object. Some good cues to boundaries between regions are depth discontinuities and concave slope discontinuities. Methods for range image segmentation are treated elsewhere [7,8,9]. In addition, the sparseness of the feature space region describing each object makes the probability of a spurious match from multiple objects low, especially for high order tripod operators. Also, in our experience, tripod operators can often detect and avoid jump boundaries because a probe curve swings out "over the cliff" and strikes the range image on the jump boundary, which is easily locally detectable. However, the experiments in this paper treat only isolated objects.

5. The Efficiency of the Tripod Operator

Most uses of the tripod operator involve the exhaustive survey of an object's surface as a preprocessing step. This is followed at recognition time by the application of the operators to a range image and the mapping the their results into the identities and/or poses of objects present. The efficiency of these steps is therefore of central interest, and they are discussed in the three succeeding sections.

5.1 How Many Ways Can a Tripod Operator Be Placed on an Object?

A tripod, when constrained to lie on a surface, can clearly move in three degrees of freedom (DOF), corresponding approximately to two translational DOF and one rotational DOF. Therefore, in order to obtain all tripod feature vector values possible from a given object, within some tolerance, one only has to densely sample a three dimensional parameter space.

We will now make essentially the same argument, in discrete terms. Suppose that the surface of an object has been tesselated into a large number m of small compact patches. Foot A of a tripod can be placed on any of the m patches. Foot B is at a fixed distance d from A, and so it can only be placed at the intersection of the object surface and a sphere of radius d centered at A. There are roughly $O(\sqrt{M})$ patches on that space curve, so there are only roughly $O(M^{3/2})$ placements possible for the first two feet. Foot C is now nearly fixed can only lie only on $O(1)$ patches. Thus $O(M^{3/2})$ is the estimated number of placements needed to exhaustively survey an object.

5.2 How Much of the Feature Space is Occupied By an Object?

Since, by the above arguments, the location of a tripod operator placement could be specified with three parameters, the resulting feature vector values must occupy at most a three dimensional manifold in feature space, regardless of its dimensionality (the order of the operator). This sparseness is useful for recognition. Of special interest are cases of surface symmetry. For example, for extruded surfaces and surfaces of revolution, sliding the operator along the symmetry direction causes no change in its feature vector. Therefore only a 2-D manifold in feature space is occupied. For a cylinder, with two symmetry directions, only a 1-D space curve in feature space is generated, resembling an ellipse. Finally, for a plane or sphere, only a single point in feature space can be obtained from any placement of a tripod operator. These properties can easily be exploited to design simple detectors for these surfaces.

5.3 How Many Surface Locations Match a Given Feature Value?

The short answer to this is "usually very few, unless the surface has special symmetries", especially for operators of order > 3. For the operators described above, the value of the feature vector uniquely determines the positions of the n probe points with respect to the 3 tripod feet. One is interested in how many ways a given object's surface could be fit to this rigid set of 3+n points by rotating and translating the object. We will give some partial answers here. We will now successively impose the constraints that each of the n+3 points lies on the surface of the object and note the effect on our knowledge of the object's pose and identity. Initially the model is free in all six degrees

of freedom (DOF). Then, as we successively require each point to lie on the model's surface, successively fewer DOF of motion are available to the model that we are trying to match to the points. That is, the set of possible poses is reduced. If no pose is possible, recognition fails for that model. Usually, introducing each additional point reduces the number of DOF by one, so that for six points (n=3), only a finite number of discrete poses are possible. If this is not the case, we say that there are *object symmetries*. For n>3, it is often the case that no object will fit except the correct one (see section 6).

Consider the symmetry cases mention in section 5.2. For a plane or sphere, an operator of order one (4 points) with a given feature value either fits everywhere on the surface or nowhere, and can can be used to recognize the surface. For a cylinder, an order 2 operator has this recognition capability, and for general extrusions, helices and surfaces of revolution, an order 3 operator does.

6. Experiments with Tripod Operators

We have implemented a software system in C on a Sun SPARCstation which allows us to perform various experiments. These involve the generation of synthetic objects, the synthesis of range images of these objects, and the application of various tripod operators to them. The experiments described here focus on measuring the ability of tripod operators to discriminate among objects, without localization.

The synthetic objects are in the form of simplicial polyhedra whose vertices lie on analytic surfaces. The choice of this class of shapes is rather arbitrary. They are nontrivial for recognition because their local surface shape is irregular. Figure 3 shows the library of such objects that we used. They were generated by a program which opportunistically fits triangles of roughly equal size to a given analytic surface, in such a way that a correct polyhedron is formed. Most of them have from 1000 to 3000 facets. The reason we went to the trouble to generate faceted surfaces is that we will need them for our future experiments in localization using linkable tripod operators and interpretation trees. Synthetic range images of these objects were generated by projecting rays through the points of a rectangular grid from a viewpoint.

6.1 Computing a Tripod Operator Placement.

The experiments in this section all involve the placement of tripod operators at random places on a range image, and so we will describe our fast procedure for doing this. We will treat the case of linkable tripod operators, such as the ones in Fig. 1d&e. We assume that a range image is given, along with formulas relating the coordinates of an arbitrary point in space with the two pixel indices of the range image. In a nutshell, the

procedure finds the intersection between a test curve and the range image by binary search along the test curve until the distance between the some point on the test curve and the corresponding range surface point is sufficiently small.

We denote the range pixel whose horizontal and vertical indices are i and j, respectively, by the 3-vector $\mathbf{r}_{ij}$. This vector is given in a coordinate system in which the viewpoint of the rangefinder is at the origin. We define $\mathbf{r}(h,v)$ as an interpolated range image such that $\mathbf{r}(h,v) = \mathbf{r}_{ij}$ if h=i and v=j. For non-integer values of h and v we will use triangulated polyhedral interpolation. Each i,j pair will yield two triangular facets; one with vertices at the range pixels (i,j), (i+1,j), and (i,j+1), and one with vertices at (i+1,j), (i,j+1), and (i+1,j+1). We denote by h(**x**) and v(**x**) the real valued functions mapping an arbitrary point **x** to the respective parameters of the corresponding point on the interpolated range image. That is, the ray from the origin of the rangefinder through **x** also passes through the range point $\mathbf{r}(h,v)$, where h = h(**x**) and v = v(**x**).

To place a tripod operator on the interpolated range image, we first place point **a** of the operator at a random range point. It may be between pixels. Then we chose a random direction in the hv plane and search for a point **b** lying on the interpolated range image at a euclidean distance d from point **a**. We do this by binary search along a circle of radius d centered at **a** for a point for a point **b** whose z component equals that of $\mathbf{r}(h(\mathbf{b}),\mathbf{v}(\mathbf{b}))$. The circle is oriented so that it is viewed edge-on from the rangefinder origin.

The third point **c** must be at a distance d from both **a** and **b**. It is calculated by binary search along a circle of radius $d\sqrt{3}/2$ centered at (**a**+**b**)/2 for a point for a point **c** whose z component equals that of $\mathbf{r}(h(\mathbf{c}),\mathbf{v}(\mathbf{c}))$. The circle is oriented coaxially with the line through **a** and **b**. Any further points in a linkable tripod operator can be computed in exactly the same way; by chosing two existing points and searching along the circle that symmetrically bisects the line segment joining them. We represent the feature values in degrees, rounded to 1^o.

Note that although there are plenty of pixels to chose from in a typical range image, the tripod operator choses only points related as described above, so that interpolated points between range pixels are often selected. We will see that this slightly awkward procedure is very well compensated for by the operator's advantages.

6.2 Discriminating Among Objects Without Explicit Models

In order to visualize point sets in feature space, we have written a program to display the first two components of the vectors resulting from randomly placing order 3

linkable tripod operators on various surfaces. Figure 4 shows some interesting examples of this. In fig. 4a we see that as discussed in section 5.2 a cylinder produces an oval space curve in feature space. Only two features suffice to measure the radius of a cylinder, since a point in the plane of Fig. 4a lies on only one oval, which corresponds to cylinders of a specific radius. Figure 4b is for our library object "tor2". Because this polyhedron is approximately a surface of revolution, having one symmetry direction, it is approximately a two dimensional region in feature space. We have visualized this (and the other examples) using a rotating computer animation of the point cloud in the 3-D feature space. It is shaped like a flower with a hole in the center. Figures 4c,d also show the reduction of a DOF due to the extrusion symmetry of a planar 90^o dihedral. This is illustrated with a planar slice. This dihedral (and other ubiquitous shapes) are good candidates for characterization by simple piecewise polynomial discriminant surfaces in feature space to enable very fast recognition.

We now describe the experiments mentioned in section 4.2. We used the order 3,4 &5 versions of the operators of Fig. 1d&e. The experiments consisted of picking an edgesize and order for the operator and a number of placements to make. Then for each object in the library, range images were generated from various viewpoints and a number of placements were made for each viewpoint, until the specified number of placements were obtained. A placement fails when one of the operator's probes either strikes a jump boundary or has no intersection with the surface. We always used 20 viewpoints, taken along the face normals of a regular dodecahedron centered on the object. This ensured likely visibility of most possible placement locations. In no way did we analyze or store aspects. The resulting set of feature vectors was stored for each object, and serves as a representation for the object.

Next, for various operator edgesize and order settings used above, new range images of some of the objects was formed, with noise added to the z component of each range pixel. The noise was obtained using a uniformly distributed pseudorandom variable in a given interval $\pm \varepsilon$. A few random placements were made, resulting in a set of feature vectors we will call a *test cloud*. The more exhaustive feature space point sets described above we call *stored clouds*, for brevity. We used these two kinds of point clouds in the experiments described below.

We focused initially on the case of order 4 operators, since the inherently 3 DOF feature space clouds should be sparse in the four dimensional space, and we wanted to use the simplest (lowest order) operator having this property. We were first concerned with how many placements it takes to "saturate" features space, so that most new

placements on the same object will be close (in feature space) to an old one. Table 1 shows the results of measuring, for each library object, the distance from each of 100 test cloud points to the nearest point in that object's previously stored cloud. The operators have edgesize .15, and there is no noise in this example. 20,000 samples were taken in generating each stored cloud, and duplicates were removed, leaving the numbers shown. Note that the near-spheres have the expected sparseness, since an ideal sphere produces only one point in any tripod operator's feature space. These two shapes completely saturate their feature space clouds with a few hundred distinct points. At the other extreme is supquad2, which retained 17,507 distinct points out of 20,000. Accordingly, only 89% of the test points are within 3^o of a stored point, whereas higher fractions were obtained for the other objects. The lesson here is that for the parameter values chosen, it is not hard to sufficiently saturate the stored cloud so that any new operator placement more than 5^o from the nearest neighbor in the cloud permits rejecting the candidate object with considerable confidence. Larger stored clouds would reduce this margin. The degree of saturation desired depends in part on the amount of noise in the range measurements. This is addressed later.

Next, we address the problem of discriminating the objects from one another. Figure 5 shows selected results of various experiments of the following form: A test cloud of size 50 was taken for a given object obj1, injecting range noise of peak magnitude ε in z, using a linkable tripod operator of given order. The operator edgesize .15 was used throughout. Then the L_2 (euclidean) distance D_2 from each test point to the nearest neighbor in the stored cloud for a different object obj2 was computed. The distance D_2 from each test point to the correct cloud (for obj1) was also computed. D_1 was plotted against D_2 for the 50 points. This presents the recognition problem in a very clear way. For example, in Fig. 5b we see that for an order 4 operator operating on a noise free range image, it is extremely easy to reject obj2 (supquad2), leaving tor1 as a candidate. Any placement having a distance to supquad2 greater than 10 (almost all of them do) immediately allows rejection of supquad2, since the likelihood of any of the objects having a gap of that size in its stored cloud is very small (recall table 1). We ran similar order 4 experiments for many pairs of objects and various values of noise.

The predominant observation in these experiments was that for most cases, the first few test points processed allowed rejection of all nine wrong objects, for realistic noise levels. The tor1 *vs* supquad2 case was chosen for Fig. 5 because it was one of the most difficult pairs to discriminate. There are probably many locally similar regions. For most other cases this typical distances to the wrong class were larger. Thus, the dominating computation was the nearest neighbor test; 20,000 distances computed for each stored

cloud, repeated several times. For this reason one of our goals is to find effective methods for speeding the nearest (or *sufficiently* near) neighbor calculation. Good candidates are binning techniques from computational geometry and interpolation methods from numerical analysis.

We will briefly digress here to discuss noise. We wish to make the point that for noise levels easily obtainable with a current rangefinder technology, remarkable recognition speed can be achieved. In our experiments we used noise of various values ranging from 0 to .015. Since our operator edgesize is d = .15 here, we used a range of ε/d of 0 to 10%. Furthermore, even for noise = .015 as in Figs. 5d,e&f, object discrimination is possible. A good triangulation rangefinder can achieve an accuracy of .5 mm, and if d = 2.5 cm, ε/d = 2%, one fifth of the Fig.5 value. This corresponds to a noise value .003 in our experiments. At this value, we were able to reject all wrong objects with the first test point most of the time.

Note that in Fig. 5 we varied order (3 values) and noise (2 values) independently. We wanted to measure the "efficiency of discrimination" e as a function of these variables. We quantified this as the fraction of the test points that would allow rejection of the wrong object (supquad2) by virtue of D_2 exceeding some reasonable threshold; namely a roughly estimated upper bound D_t on the distance D_1 to the correct class. D_t was obtained from data similar to that of table 1. D_t depends on operator order and on noise. We obtained these results for the Fig. 5 data:

a: $D_t = 4$, $e = 12/50$.
b: $D_t = 5$, $e = 19/50$.
c: $D_t = 7$, $e = 38/50$.
d: $D_t = 10$, $e = 6/50$.
e: $D_t = 15$, $e = 5/50$.
f: $D_t = 20$, $e = 3/50$.

We observed that as order was increased in a through c without noise, the efficiency rapidly increased in response to the increased feature information. However, as order increased in d through f in the presence of substantial range noise, the efficiency seems to have decreased somewhat, probably due to the fact that noise *in feature space distance* increases with the number of components of the feature vector and counteracts the information gain.

7. Conclusions and Future Directions

We have introduced a new class of operators for range images and presented theoretical and experimental evidence of their usefulness in recognizing previously known rigid objects in range images. Tripod operators were shown to generate manifolds of dimensionality not exceeding three in feature space of any order. They provide a way to measure in constant time the distinctiveness of a local region of a range image, in terms of both the number of models eliminated and the number of placements on the models eliminated. Our experiments on fast rejection of a number of objects were extremely successful and show that with reasonably accurate rangefinders many objects can often be rejected with a single operator placement. This is possible because a few (e.g. 7) range measurements contain much intrinsic shape information about a surface, and tripod operators separate this information from pose information completely.

Tripod operators make the localization problem amenable at least partly to treatment by lookup tables, and are highly compatible with the method of constrained search of interpretation trees, allowing the use of other constraints along with the tripods.

These operators suggest a great variety of future work. Their use in a complete vision system needs to be studied experimentally. Various traditional statistical pattern recognition methods might be useful for improving the model-free classification approach studied here, since tripod operators generate low-dimensional, highly informative feature vectors. For example, a torus-like discriminate surface in a 3-D feature space could detect a cylinder. For higher order feature spaces than three, lookup tables are not even feasible, and analytic approximations of the 3-D subspaces for various objects might be very effective. This could lead to extremely fast recognition by eliminating the nearest neighbor search. Also, mechanical tactile tripod operators might enable very fast tactile recognition.

Some flexible objects might be recognizable with some variant of the tripod operator, since when linked via three points they enforce local shape constraints more strongly than global ones, thus providing a potential method of approximating the continuum mechanics of bending an object.

In the near future, we plan to generate for various tripod operators, modeled objects, and amounts of noise the set of possible interpretations consistent with each value of the feature vector for that operator. This will then allow us to better answer such questions as how accurate a rangefinder is required for various recognition problems, what kind of tripod operators are most useful, how fine a surface tessellation is required in the model, and what speedup over the pure interpretation tree approach is provided. We will study these problems in the context of building a high performance prototype recognition system.

References

[1] Grimson, W. E. L., The Combinatorics of Object Recognition in Cluttered Environments Using Constrained Search, MIT AI Memo No. 1019, February, 1988.

[2] Gaston,P. C., and Lozano-Perez, T., Tactile Recognition and Localization Using Object Models: The Case of Polyhedra On A Plane, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6 (3):257-265, May 1984.

[3] Oshima, M. and Shirai, Y., Object Recognition Using Three-Dimensional Information, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-5(4):353-361, July, 1983.

[4] Faugeras, O.D., and Hebert, M., A 3-D recognition and Positioning Algorithm Using Geometrical Matching Between Primitive Surfaces, Proc. Eighth Int. Joint Conf. Artificial Intelligence, pp 996-1002, August, 1983.

[5] Grimson, W.E.L., and Lozano-Perez, T., Model-Based Recognition and Localization from Sparse Range or Tactile Data, The International Journal of Robotics Research, Vol. 3, No. 3, pp 3-35, Fall 1984.

[6] Rioux, M., Blais, F., Beraldin, J., and Boulanger, P., Range Imaging Sensors Development at NRC Laboratories, Proc. of the Workshop on Interpretation of 3-D Scenes, pp 154-160, Nov. 27, 1989, Austin, TX, IEEE Press.

[7] Yokoya, N., and Levine, M.D., Range Image Segmentation Geometry Based on Differential Geometry: a Hybrid Approach, IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-11 (6):643-649, July, 1983.

[8] Faugeras, O.D., Hebert, M, and Pauchon, E., Segmentation of Range Data into Planar and Quadratic Patches, in Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp 8-13, June, 1983.

[9] Han, J.H., and Volz, R.A., Region Grouping from a Range Image, Proc. IEEE Conf. on Computer Vision and Pattern Recognition, pp241-248, June 1988, IEEE Press.

[10] Bolles, R.C., and Cain, R.A., Recognizing and Locating Partially Visible Objects: The Local-Feature-Focus Method, International Journal of Robotics Research 1(3):57-82.

[11] Lamdan, Y., and Wolfson, H.J., "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme", IEEE 2nd International Conf. on Computer Vision, 1988.

[12] Pipitone, F., "Tripod Operators for the Interpretation of Range Images", NRL Memorandum Report #6780, February, 1991.